REMARKS

Claims 1-27 are pending and under consideration. Reconsideration is requested based on the foregoing amendment and the following remarks.

Objections to the Drawings:

The drawings were objected to for including terms that are asserted to be unclear, such as "linker" and "load module." The terms "linker" and "load module," are, to the contrary, submitted to be clear within the requirements of 35 U.S.C. § 112, first paragraph, as discussed more fully below. Since there is no way to correct the drawings to define the meaning of the terms "linker" and "load module" more fully, short of inserting the definitions of the terms "linker" and "load module" from the specification into the drawings, no replacement drawings are being submitted. Withdrawal of the objections to the drawings is earnestly solicited.

Objections to the Specification:

The Title was objected to for not being descriptive. The Title has been amended to make it more descriptive. Withdrawal of the objection is earnestly solicited.

Objection to the Abstract:

The Abstract was objected to for various informalities. Appropriate corrections were made. Withdrawal of the objection is earnestly solicited.

Objections to the Specification under 35 U.S.C. § 112, first paragraph:

The Specification was objected to under 35 U.S.C. § 112, first paragraph, as not being written in "full, clear, concise, and exact terms." The first paragraph of 35 U.S.C. § 112 provides,

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Thus, under the provisions of 35 U.S.C. § 112, first paragraph, a written description need not be so full, clear, concise, and exact as to enable *any* person to make and use the invention, rather, the written description need only enable any person *skilled in the art* to make and use the invention.

The Office Action, on the other hand, seems to be laboring under the perception that the specification must be written so that *any* person, not just any person *skilled* in the art, will be able to understand it. In particular, the Office Action asserts that the terms "load module" and "linker" are not defined. To the contrary, a flow chart depicting steps that may be taken to create a load module appears in Fig. 5. According to Fig. 5, as well as the description accompanying Fig. 5 in the specification at page 3, lines 8-14,

> FIG. 5 is a flowchart that explains process steps of a load-module creation for the sample program shown in FIG. 4. First, a source code of the program is converted into an assembly code using a compiler (steps S501 to S503). An object is created from the assembly code using an assembler (steps S504 to S506). Plural objects are linked using a linker to create a load module for the program (steps S507 to S510).

Thus a load module is a collection of object modules, linked together, that may also be linked to, for example, libraries and data, for execution on a computer. The linker links them together.

A programming language may have, for example, libraries of canned subroutines available for all users. A user may simply invoke a particular subroutine from the library when writing the program by inserting the name of the subroutine and the appropriate arguments. In order to execute the program, however, the computer must have the code supporting the canned subroutine, the name of the subroutine by itself will not do. The code for the subroutine contained in a library may, in that circumstance, be made available to the program by the linker while the load module is being created. The linker essentially links the main program to various other sources of data, such as libraries.

Furthermore, implementation of a load module is shown in Fig. 17. In particular, as described at page 23, lines 22-25, continuing at page 24, lines 1-6,

> FIG. 17 is a block diagram that shows the functional structure of the load-module creating apparatus according to the first embodiment. The functions of each sections are implemented by programs stored in the hard disk 1605, and floppy disk 1607 shown in FIG. 16 and read into the random-access memory 1603 by the central processing-unit 1601. The programs are, in essence, a compiler, an assembler, and a linker.
> The functions of sections 1700 to 1704 are implemented by the compiler, the functions of sections 1705 to 1707 are implemented by the assembler, and the functions of sections 1708 to 1712 are implemented by the linker.

Creation of a load module is shown in Fig. 18. In particular, as described at page 24, lines 11-25, continuing at page 25, lines 1-6,

> In the load-module creating apparatus, first the compiler is activated. A first analyzing section 1700 implemented by the compiler reads a source code of a

specified program, carries out lexical analysis and parsing of the source code, and converts the program into an internal representation of the compiler (step S1801).
Next, a shared data determining section 1701 determines, by scanning the internal representation of the compiler, if individual data included therein is shared among the different processes of the program. When a data is determined to be a shared data, an identifier to indicate that the data is shared is affixed to the data (step S1802).
Next, a shared data identification information affixing section 1702 scans the internal representation of the compiler, finds the data with the identifier affixed in step S1802, and for all the shared data thus found, affixes as an identification information a prefix to the data name (step S1803). The prefix may for instance be `_shr_`.
Next, an instruction string creating section 1703 creates, based on the internal representation of the compiler, instruction strings that run the program and adds the instruction strings to the internal information of the compiler (step S1804).

Thus, the term "load module" is submitted to be clear to a person skilled in the art, within the meaning of 35 U.S.C. § 112, first paragraph.

The activities of the linker are described at page 25, lines 24 and 25, continuing at page 26, lines 1-20,

Next, the linker is activated in the load-module creating apparatus. An object reading section 1708 implemented by the linker reads as an internal representation of the linker the object output by the object output section of the assembler in step S1808 (step S1809).
Next, a shared data area forming section 1709 searches the internal representation of the linker for data with the shared data identification information (such as the prefix shr), forms an area (shared data area 1102 in FIG. 11) that includes only shared data, and adds it as the internal representation of the linker (step S1810).
Next, a memory space building section 1710 creates an area (data area 1101 in FIG. 11) that includes only the private data that is left behind in the internal representation of the linker and another area (text area 1100 in FIG. 11) that includes only the instruction strings (step S1811).
Next, an address resolving section 1711 carries out resolution of the address of each of the memory areas, namely, the text area 1100, the data area 1101 and the shared data area 1102 of the internal representation of the linker (step S1812).
Next, a load module output section 1712 outputs, based on the internal representation of the linker, a load module for the program (step S1813). This completes the assembling of the object code by the linker and the conversion of source code to the load module.

Thus, the term "linker" is submitted to be clear to a person skilled in the art, within the meaning of 35 U.S.C. § 112, first paragraph.

The term "affixing" is a common English word meaning, e.g. "attaching," or "adhering," or "appending," or, in this case, "associating with." There are not two definitions of the term "affixing" used in the claims, contrary to the assertion in the Office Action. Rather, there are two different objects that are the subject of the affixing, i.e. "a specific prefix" and "an invalidate instruction" are to be affixed under different circumstances. Thus, the term "affixing" is submitted to be clear to a person skilled in the art, within the meaning of 35 U.S.C. § 112, first paragraph.

The term "uncached" has been replaced with the term "prevented from being cached" in the Abstract, the only place it appeared.

The term "symbol" is another common English word meaning, e.g. "representative of another object." Thus, the term "symbol" is submitted to be clear to a person skilled in the art, within the meaning of 35 U.S.C. § 112, first paragraph.

The specification is thus submitted to meet the requirements of 35 U.S.C. § 112, first paragraph. Withdrawal of the objections is earnestly solicited.

Claim Rejections - 35 U.S.C. § 112:

Claims 1-27 were rejected under 35 U.S.C. § 112, first paragraph, for failing to comply with the enablement requirement. The Office Action asserts that claims 1-27 are not enabled for essentially those reasons discussed above with respect to the objections to the specification. Claims 1-27 are consequently submitted to be enabled within the requirements of 35 U.S.C. § 112, first paragraph, for essentially those reasons discussed above. If a telephone or office consultation would help resolve this issue, the courtesy of a telephone call to the undersigned representative of the Applicants is requested. Otherwise, withdrawal of the rejection is earnestly solicited.

Rejections under 35 U.S.C. § 112, second paragraph:

Claims 1-27 were rejected under 35 U.S.C. § 112, second paragraph, as indefinite. The Office Action asserts that claims 1-27 are indefinite for essentially those reasons discussed above with respect to the objections to the specification. Claims 1-27 are, consequently submitted to be definite within the requirements of 35 U.S.C. § 112, second paragraph, for essentially those reasons discussed above.

Furthermore, the term "uncached" does not appear in the claims. The term "non-cacheable area" appearing in the claims is described at page 16, lines 20-25, continuing at page 17, line 1.

By somewhat diverting this function, a non-cacheable area can be provided in the memory space of the processor and the shared data can be limited only to this area. By doing this, it can be ensured that only data related to the part of the program being executed by each processor, that is, private data, remains in the cache memory and that since shared memory does not exist in the cache, always the main memory is accessed for reading/writing values.

The term "referred" does not appear in the claims. Where it appears in the Abstract, the preposition "to" has been added after it for purposes of clarification.

The term "specific expression" does not appear in the claims.

The terms "load module," "affixing," and "symbol" were discussed above with respect to the objections to the specification, and are thus submitted to be definite within the requirements of 35 U.S.C. § 112, second paragraph, for those reasons discussed above.

Claims 1-27 are consequently submitted to be definite within the requirements of 35 U.S.C. § 112, second paragraph. If a telephone or office consultation would help resolve this issue, the courtesy of a telephone call to the undersigned representative of the Applicants is requested. Otherwise, withdrawal of the rejection is earnestly solicited.

Conclusion:

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 30 Sep 05

By [illegible]
Thomas E. McKiernan
Registration No. 37,889

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501